C H A P T E R 9

PCI

This chapter presents the PC 99 requirements and recommendations for Peripheral Component Interconnect (PCI) host controllers and peripherals.

The PCI architecture has become the most common method used to extend PCs for add-on adapters. Windows and Windows NT use the basic PCI infrastructure to gain information about devices attached to the PCI bus. The ability of PCI to supply such information makes it an integral part of the Plug and Play architecture in Windows.

[Note to Reviewers: Processor and core chip set technologies are moving to lower operating voltages, enabling increased speeds and lower power consumption. These advances will result in systems that require 3.3V signalling for PCI (as opposed to 5V signalling that is used today). To make this transition as smooth as possible, two items (5 and 6) are defined. Item 5 requires PCI systems to provide 3.3V power to PCI connectors. This enables the cost effective development of 3.3V PCI adapters. Item 6 recommends that PCI add-on devices be 'Universal Boards' which will operate in either 5V or 3.3V signalling environments. See Section 4.1 of the PCI Specification (Rev 2.1 or Rev 2.2) for a description of the 5V to 3.3V signalling transition and Universal Boards.

Specific requirements related to PCI are defined in the following chapters:

- Requirements for dual IDE adapters that use PCI are defined in the "ATA and ATAPI" chapter in Part 3 of this guide.
- Requirements for PCI-to-PC Card bridges are defined in the "PC Card" chapter in Part 3 of this guide.
- Requirements for graphics devices that use PCI are defined in the "Graphics Adapters" chapter in Part 4 of this guide.
- Requirements for audio implementations that use PCI are defined in the "Audio Components" chapter in Part 4 of this guide.

Published: April 24, 1997. Printed: 04/24/98 9:52 AM

Contents

PCI Basic Requirements

This section summarizes the basic design requirements for PCI.

1. All components comply with PCI 2.1
Required

Recommended: All components comply with PCI 2.2.

All cards, bridges, and devices that use PCI ~~must~~ should be designed to meet the requirements defined in *PCI Local Bus Specification, Revision 2.2* (PCI 2.2), and must be designed to meet the requirements defined in *PCI Local Bus Specification, Revision 2.1* (PCI 2.1). PCI specifications are available from the PCI SIG, with information available at http://www.pcisig.com/.

Compliance with this requirement is demonstrated based on the compliance process of the PCI Special Interest Group (SIG).

2. System does not contain ghost cards
Required

A computer must not include any ghost cards, which are cards that do not decode the type 1/type 0 indicator. Such a card will appear on bus 0 as all the buses behind it that use the same IDSEL. Notice that it is acceptable, as defined in PCI 2.1 or later, for a single-function card to decode the IDSEL and AD[1::0] pins and not decode AD[10::08] if the card does not have bit 7 set in the header type. This requirement also excludes, for example, devices that ignore some type 0 transaction bits and therefore appear at multiple device/function addresses.

A PCI card should be visible through hardware configuration access at only one bus/device/function coordinate.

3. System uses standard method to close BAR windows on nonsubtractive decode PCI bridges
Required

PCI-to-PCI bridges must comply with the *PCI to PCI Bridge Specification, Revision 1.0.* Setting the base address register (BAR) to its maximum value and the limit register to zeros should effectively close the I/O or memory window references in that bridge BAR.

Published: April 24, 1997. Printed: 04/24/98 9:52 AM

4. System supports PCI docking through a bridge connector
Recommended

It is recommended that the system support docking through a bridge connector, with the actual bridge on the docking station, not on the mobile unit. The bridge can be positive or subtractive decoding. The bridge should create a new bus number so devices behind the bridge are not on the same bus number as other devices in the system.

After a warm dock, the BIOS should not configure the bridge or any other devices in the docking station. That is the responsibility of the operating system.

Mobile PC Note

The PCI-to-ISA bridge should be placed on the docking station, not on the mobile unit. Mobile PCs typically do not have ISA expansion slots, and the ISA devices on the mobile PC can be controlled by the Plug and Play interface. For more information on requirements for docking station systems, see the "PC Basic Requirements" chapter in Part 2 of this guide.

Notice that implementing delayed transactions for PCI-to-PCI and PCI-to-ISA docking bridges is required in PCI 2.1 or later only when certain timing conditions are not met. For PC 99 design requirements, this is interpreted to mean that delayed transactions are required only when "targets cannot complete the initial data phase within the requirements of this specification" (as stated in PCI 2.1 or later). Delayed transactions are a hardware-related timing issue (and will provide a performance advantage), but are not related to operating system requirements.

5. System provides 3.3V to all PCI connectors
Required

PC 99 systems are required to provide 3.3V (with adequate amperage) to all PCI connectors. This requirement enables the development of 3.3V PCI adapters without the cost of a voltage regulators.

6. PCI add-on devices support both 5V and 3.3V signalling
Recommended

All PCI add-on devices should be 'Universal Boards' as defined in Section 4.1 of the PCI Specification Revision 2.1 or 2.2. These Universal Boards support both 3.3V and 5V signalling, and will allow a smooth market transition as systems change from 5V to 3.3V signalling. This item will be required in future versions of this design guide.

PCI Controller Requirements

This section summarizes PCI controller requirements.

7. System-board bus complies with PCI 2.1
Required

Recommended: System-board bus complies with PCI 2.2.

Published: April 24, 1997. Printed: 04/24/98 9:52 AM

The system-board bus hardware should~~must~~ comply with PCI 2.2 and must comply with PCI 2.1. The bus design must fully implement all bus requirements on every expansion card connector.

8. Bus master privileges are supported for all connectors
Required

To ensure full Plug and Play functionality on a PCI bus with expansion cards, all PCI connectors on the system board must be able to allow any PCI expansion card to have bus master privileges.

9. Functions in a multifunction PCI device do not share writable PCI Configuration Space bits
Required

The operating system treats each function of a multifunction PCI device as an independent device. As such, there can be no sharing between functions of writable PCI Configuration Space bits (such as the Command register).

Notice that the PC Card 16-bit Interface Legacy Mode BAR—offset 44h in the Type 2 PCI header—is the only exception to this requirement. This register must be shared between the two functions, just as they must share the same compatibility registers with the Exchangeable Card Architecture (ExCA) programming model, as defined in the *PCI to PCMCIA CardBus Bridge Register Description* (Yenta specification), by Intel.

For more information about design requirements for CardBus controllers, see the "PC Card" chapter in Part 3 of this guide.

10. All PCI devices complete memory write transaction (as a target) ~~within 10 microseconds~~within specified times~~.~~
Required

All devices must comply with the Maximum Completion Time ECN that is approved for the PCI 2.1 specification. This requirement is also documented in the PCI 2.2 specification (section TBD). Compliance with this requirement provides shorter transaction latencies on PCI, allowing more robust handling of isochronous streams in the system.

Plug and Play for PCI Controllers and Peripherals

This section summarizes the Plug and Play requirements for PCI devices.

11. Devices use PCI 2.1~~2~~ Configuration Space register for Plug and Play device ID
Required

The PCI specification describes the Configuration Space register used by the system to identify and configure each device attached to the bus. The Configuration Space is made up of a 256-byte field for each device and contains

Published: April 24, 1997. Printed: 04/24/98 9:52 AM

sufficient information for the system to identify the capabilities of the device. Configuration of the device is also controlled from this register.

The Configuration Space register is made up of a header region and a device-dependent region. Each Configuration Space register must have a 64-byte header at offset 0. All the device registers that the device circuit uses for initialization, configuration, and catastrophic error handling must fit in the space between byte 64 and byte 255.

All other registers that the device uses during normal operation must be located in normal I/O or memory space. Unimplemented registers or reads to reserved registers must complete normally and return zero (0). Writes to reserved registers must complete normally, and the data must be discarded.

12. Device IDs include Subsystem IDs

Required

The Subsystem ID and Subsystem Vendor ID fields are required for compliance with the Subsystem ID ECN to PCI 2.1, PCI 2.2, and PC 99. Support for these registers requires non-zero values to be populated for both the Subsystem ID and Subsystem Vendor ID.

These fields are necessary for the correct enumeration of a device. When the Subsystem ID fields are populated correctly for the adapter, Windows can differentiate between adapters based on the same PCI chip.

The Subsystem ID also allows Windows to load system miniports for system-board devices. Thus, Subsystem IDs are also a requirement on system-board devices. The exceptions to this requirement are PCI-to-PCI bridges and core chip sets.

PCI ~~2.2~~ and PC 99 require that all PCI functions ensure that the Subsystem IDs are in place before the operating system accesses the function's configuration space registers. This is required both at initial operating system load and after any transition of the PCI bus from B3 (unpowered) back to B0.

PCI ~~2.2~~ and PC 99 also require that this be done by hardware on the card itself—for example, by way of serial EEPROM—and not by an extension BIOS or device driver. This is because the code is not guaranteed to run in all relevant cases, especially system sleep or dynamic bus power state transitions in which the bus becomes unpowered.

It is acceptable for the system BIOS (at POST) or ACPI control methods (at run time for sleep, bus power state transitions, or both) to program the Subsystem IDs. This code is always run before the operating system accesses a function's configuration space.

Hardware to support this requirement can be designed in any of three ways:

Published: April 24, 1997. Printed: 04/24/98 9:52 AM

- Load the value by hardware methods—for example, pin strapping at RST, an attached parallel or serial ROM, and so on.
- Make a copy of the Subsystem ID field and the Subsystem Vendor ID field in PCI user-defined space. Any writes to these locations will change both the copy and the original field. Any writes to the original field are discarded. POST code or control methods access the copy fields.
- Make a write-enable bit in the PCI user-defined space. The POST code or control method can turn this bit on, change the Subsystem Vendor ID, and then turn it off.

13. Configuration Space is correctly populated
Required

Windows places extra constraints on a few configuration registers and has uncovered some problem usage of other registers. Microsoft provides a program named Pci.exe to help debug the use of the Configuration Space. This program is available on the Microsoft FTP server, as described in the "PCI References" section at the end of this chapter.

The following items are specific requirements for the Configuration Space:

- Populate the class code register (09h) for all devices.

  Follow the base class, sub-class, and programming interface values outlined in PCI 2.1 or later.
- Devices must not fill BARs with random values.

  See PCI 2.1 or later for correct usage of these registers. Notice that BARs (10, 14, 18, 1C, 20, and 24h) should return zero if they are not used, indicating that no memory or I/O space is needed.

Also, for performance reasons, it is recommended that run-time registers for PCI devices should not be placed in the Configuration Space.

14. Interrupt routing supported using ACPI
Required

The system must provide interrupt routing information using a _PRT object, as defined in Section 6.2.3 of the *Advanced Configuration and Power Interface Specification, Revision 1.0* or later.

15. BIOS does not configure I/O systems to share PCI interrupts
Recommended

This applies to boot devices configured by the BIOS on systems based on Intel Architecture processors. The operating system should configure all other devices. For systems that will run the Windows operating system, OEMs should design the BIOS so that it does not configure the I/O systems in the PC to share PCI interrupts for boot devices. An exception exists for legacy audio devices following the configuration guidelines outlined in the white paper titled

Published: April 24, 1997. Printed: 04/24/98 9:52 AM

Implementing Legacy Audio Devices on the PCI Bus, available on the web site at http://www.intel.com/pc-supp/platform/ac97/wp/leg_pci.htm.

Windows does not support sharing an IRQ between real-mode and protected-mode code within the I/O subsystem. An example of this is an NDIS 2.0 driver (real mode) and a SCSI miniport driver (protected mode) for two PCI devices that share the same IRQ. The problem is that the IRQ needs to be reflected to real mode for the NDIS 2.0 driver to work.

However, if the IRQ is reflected to real mode, the real-mode SCSI driver (which usually is not called because Windows takes over in protected mode) might touch the hardware, which would cause the SCSI miniport to be confused. Windows resolves this problem either by switching everything to protected mode or by falling back to real mode.

16. BIOS configures boot device IRQ and writes to the interrupt line register
Required

This requirement applies to boot devices configured by the BIOS on systems based on Intel Architecture processors. Windows should configure all other devices because, after an IRQ is assigned by the system BIOS, Windows cannot change the IRQ, even if necessary. If the BIOS assigns the IRQ and Windows needs it for another device, a sharing problem occurs.

The BIOS must configure the boot device IRQ to a PCI-based IRQ and must write the IRQ into the interrupt line register 3Ch, even if the BIOS does not enable the device. This way, the operating system can still enable the device with the known IRQ at configuration time, if possible.

17. Systems supporting hot swapping for any PCI device use ACPI-based methods
Required

Windows 98 and Windows NT 5.0 support dynamic enumeration, installation, and removal of PCI devices only if there is a supported hardware insert/remove notification mechanism.

The appropriate notification mechanism is supported as a bus standard for CardBus bus controllers. For other solutions, such as those required for docking stations or hot-plug PCI devices, the hardware insert/remove notification mechanism must be implemented as defined in Section 5.6.3 of the ACPI 1.0 specification.

Power Management for PCI Controllers and Peripherals

This section summarizes the specific PCI power management requirements.

Published: April 24, 1997. Printed: 04/24/98 9:52 AM

18. All PCI components comply with PCI Bus Power Management Interface specification

Required

The PCI bus, any PCI-to-PCI bridges on the bus, and all devices on the PCI bus must comply with *PCI Bus Power Management Interface Specification, Revision 1.1* or later.

For PC 99 systems, PCI bus implementations must support all four device power states, bus power states (as summarized later in this requirement), and the standard PME# and 3.3 Vaux wake-up mechanisms.

System support for delivery of 3.3 Vaux to the PCI bus must be capable of powering a single PCI slot with 375 mA at 3.3 Vaux, with 20 mA at 3.3 Vaux for each of the other PCI slots on the segment whenever the PCI bus is in the B3 state.

Systems must be capable of delivering 375 mA at 3.3 Vaux to all PCI slots whenever the PCI bus is in any "bus powered" state—that is, B0, B1, or B2.

Support for bus power states can be implemented as follows:

- **System Sleeping Case (S1, S2, S3, S4):** PCI bus must be in a bus state (B*x*) no higher than the system sleeping state (S*x*). This means that if the system enters S1, the bus must be in B1, B2, or B3.

  If the system enters S2, the bus must be in B2 or B3, and if the system enters S3, the bus must be in B3. Of course, in S4 and S5, the system power is removed, so the bus state is B3. Transition of the PCI bus power states can be coupled to the sleep/wake logic.

- **System Working Case (S0):** Bus power states are optional. If implemented, there must be software-accessible controls that can be used by the operating system to control the PCI bus segment's power state dynamically, as device power states change. This control is provided at the bus segment's originating bus bridge.

  For CPU-to-PCI bridges, these controls must be implemented using ACPI or the PCI Power Management Interface standard. For PCI-to-PCI bridges, these controls must be implemented in compliance with the *PCI Power Management Interface* specification. Transition of the PCI Bus power states cannot be coupled to the sleep/wake logic.

For all devices except core chip-set components and docking bridges, compliance with the *PCI Bus Power Management Interface* specification is required, including the Configuration Space registers and the device state (D*x*) definitions. ACPI is *not* an acceptable alternative for PC 99.

Published: April 24, 1997. Printed: 04/24/98 9:52 AM

PCI-based modem and network adapters have additional requirements:

- Modem adapters must be capable of generating a power management event (PME# assertion) from the D3 cold device state. Modem adapters also must support capture of Caller ID with hardware support for the AT+VRID, "resend caller ID", voice modem command.
- Network adapters must support the generation of a power management event (PME# assertion) from the D3 cold device state. Network adapters also must support network packet filtering capabilities as defined in the *Network Device Class Power Management Reference Specification, Version 1.0* or later.

PCI References

The following represents some of the references, services, and tools available to help build hardware that is optimized to work with Windows operating systems.

Advanced Configuration and Power Interface Specification, Revision 1.0
http://www.teleport.com/~acpi/

"Efficient Use of PCI," Platform Architecture Labs, Intel Corporation
http://support.intel.com/oem_developer/chipsets/pci/general/pci001.htm

"IDs and Serial Numbers for Plug and Play" and other related articles
http://www.microsoft.com/hwdev/busbios/

Implementing Legacy Audio Devices on the PCI Bus
http://www.intel.com/pc-supp/platform/ac97/wp/leg_pci.htm

Microsoft testing tools, specifications, and information
E-mail: pciinfo@microsoft.com
http://www.microsoft.com/hwtest/
http://www.microsoft.com/hwdev/busbios/
ftp://ftp.microsoft.com/developr/drg/plug-and-play/pci/pci.exe

PCI Bus Power Management Interface Specification, Revision 1.1
PCI Local Bus Specification, Revision 2.1 and later
PCI to PCI Bridge Specification, Revision1.0.
PCI SIG
Phone: (800) 433-5177
http://www.pcisig.com/

PCI to PCMCIA CardBus Bridge Register Description (Yenta specification)
PCMCIA
2635 North First Street, Suite 209
San Jose, CA 95134 USA
Phone: (408) 433-2273
Fax: (408) 433-9558
E-mail: office@pcmcia.org
http://www.pc-card.com/

Published: April 24, 1997. Printed: 04/24/98 9:52 AM

Checklist for PCI

If a recommended feature is implemented, it must meet the PC 99 requirements for that feature as defined in this document.

1. All components comply with PCI 2.1
Required

2. System does not contain ghost cards
Required

3. System uses standard method to close BAR windows on nonsubtractive decode PCI bridges
Required

4. System supports PCI docking through a bridge connector
Recommended

5. System provides 3.3V to all PCI connectors
Required

6. PCI add-on devices support both 5V and 3.3V signalling
Recommended

7. System-board bus complies with PCI 2.1
Required

8. Bus master privileges are supported for all connectors
Required

9. Functions in a multifunction PCI device do not share writable PCI Configuration Space bits
Required

10. All PCI devices complete memory write transaction (as a target) within specified times
Required

11. Devices use PCI 2.1 Configuration Space register for Plug and Play device ID
Required

12. Device IDs include Subsystem IDs
Required

13. Configuration Space is correctly populated
Required

14. Interrupt routing supported using ACPI
Required

15. BIOS does not configure I/O systems to share PCI interrupts
Recommended

16. BIOS configures boot device IRQ and writes to the interrupt line register
Required

17. Systems supporting hot swapping for any PCI device use ACPI-based methods
Required

18. All PCI components comply with PCI Bus Power Management Interface specification
Required

Published: April 24, 1997. Printed: 04/24/98 9:52 AM